

FS4412 mjpg-streamer 视频流服务器移植文档

华清远见研发中心

我们需要做一个基于 FS4412 的视频服务器，于是 mjpg-streamer 这个开源软件变成了我们的选择之一，并且成为我们最终的选择，mjpg-streamer 提供多种输入及输出方式。输入可以是一个 camera 也可以是一个文件，输出可以是一个文件也可以以 HTTP 的方式 输出到远程客户端，这里我们选择的方式是从 camera 中取数据以 HTTP 的方式发送到远端客户端。远端客户端可以是任何支持 stream 传输的 web 浏览器。

1、JPEG 库移植

FS4412 配备摄像头 OV3640，且通过 FIMC 去操作摄像头，最终从提取出的数据格式是可以由用户控制这里我们选择比较简单的一种格式 YUYV，YUYV 数据如果是想保存为图片，或进行网络传输显示到网页上，则需要将一帧图像格式化为 mjpg 的格式，所以我们需要移植 jpeg 的库。

a) 下载 jpeg 库

通过下面这个网址下载 jpeg 源码：<http://www.ijg.org/files/jpegsrc.v8b.tar.gz>

b) 配置 jpeg 源码

```
$ tar xvf jpegsrc.v8b.tar.gz
$ cd jpeg-8b
$ ./configure --prefix=/home/david/Exynos4412/test/mjpg-streamer/jpeg
--host=arm-cortex_a8-linux-gnueabi
```

--prefix 用来指定编译后软件的安装路径；

--host 用来指定编译时使用的工具链；

这两个参数 可以根据自己的实际情况来修改。

配置成功显示如下：

```
checking whether the arm-none-linux-gnueabi-gcc -std=gnu99 linker
-lc (arm-none-linux-gnueabi/bin/ld) supports shared libraries... yes
checking whether -lc should be explicitly linked in... no
checking dynamic linker characteristics... GNU/Linux ld.so
checking how to hardcode library paths into programs... immediate
checking whether stripping libraries is possible... yes
checking if libtool supports shared libraries... yes
checking whether to build shared libraries... yes
checking whether to build static libraries... yes
checking libjpeg version number... 8:2
configure: creating ./config.status
config.status: creating Makefile
config.status: creating jconfig.h
config.status: executing depfiles commands
config.status: executing libtool commands
```

c) 编译和安装

```
make
```

```
make install
```

编译成功显示如下:

```
CC      wrbmp.o
CC      rdcolmap.o
CCLD    djpeg
CC      jpegtran.o
CC      transupp.o
CCLD    jpegtran
CC      rdjpgcom.o
CCLD    rdjpgcom
CC      wrjpgcom.o
CCLD    wrjpgcom
make[1]: Leaving directory `/home/david/Exynos4412/test/mjpg-streamer/jpeg-8b'
```

安装成功显示如下:

```
mjpg-streamer/jpeg/bin/wrjpgcom
/bin/bash /home/david/Exynos4412/test/mjpg-streamer/jpeg-8b/install-sh -
/home/david/Exynos4412/test/mjpg-streamer/jpeg/include
/usr/bin/install -c -m 644 jconfig.h /home/david/Exynos4412/test/mjpg-st
mer/jpeg/include/jconfig.h
test -z "/home/david/Exynos4412/test/mjpg-streamer/jpeg/include" || /bin
dir -p "/home/david/Exynos4412/test/mjpg-streamer/jpeg/include"
/usr/bin/install -c -m 644 jerror.h jmorecfg.h jpeglib.h '/home/david/E
os4412/test/mjpg-streamer/jpeg/include'
test -z "/home/david/Exynos4412/test/mjpg-streamer/jpeg/share/man/man1"
/bin/mkdir -p "/home/david/Exynos4412/test/mjpg-streamer/jpeg/share/man/
1"
/usr/bin/install -c -m 644 cjpeg.1 djpeg.1 jpegtran.1 rdjpgcom.1 wrjpgc
1 '/home/david/Exynos4412/test/mjpg-streamer/jpeg/share/man/man1'
make[1]: Leaving directory `/home/david/Exynos4412/test/mjpg-streamer/jp
8b'
```

查看安装路径/home/david/Exynos4412/test/mjpg-streamer/jpeg:

```
david@ubuntu:~/Exynos4412/test/mjpg-streamer/jpeg$ ls
bin  include  lib  share
```

- d) 将编译好的库文件拷贝到根文件系统中

```
cp /home/david/Exynos4412/test/mjpg-streamer/jpeg/lib/* /source/rootfs/lib
```

2、mjpg-streamer 移植

- a) 下载 mjpg-streamer 源码:

通过下面地址下载 mjpg-streamer 源码包:

<http://sourceforge.net/projects/mjpg-streamer/>

- b) 配置 mjpg-streamer 源码:

修改 Makefile 文件

修改顶层目录及所有子目录中的 Makefile 文件:

修改

```
CC = gcc
```

为

```
CC = arm-cortex_a8-linux-gnueabi-gcc
```

再次修改 plugin/input_uvc/Makefile 文件

在第 16 行添加编译时需要的 jpeg 头文件的路径

```
CFLAGS += -I /home/david/Exynos4412/test/mjpg-streamer/jpeg/include
```

在第 24 行添加编译时需要的 jpeg 库的路径

修改

```
$(CC) $(CFLAGS) -ljpeg -o $@ input_uvc.c v4l2uvc.lo jpeg_utils.lo dynctrl.lo
```

为

```
$(CC) $(CFLAGS) -ljpeg -L /home/david/Exynos4412/test/mjpg-streamer/jpeg/lib -o $@  
input_uvc.c v4l2uvc.lo jpeg_utils.lo dynctrl.lo
```

修改 plugins/input_uvc/v4l2uvc.c

这个目录是支持 uvc 摄像头的，而我们的 FS4412 的摄像头使用的 samsung 的 FIMC 架构，所有在应用程序编程时有点细微差别，这里需要针对这些差别做一些改动：

修改 static int init_v4l2(struct v4l2 *vd)函数

在

```
/*
```

```
 * set format int
```

```
*/
```

前添加如下代码

```
int input = 0;
```

```
ret = ioctl(vd->fd, VIDIOC_S_INPUT, &input);
```

```
if (ret < 0) {
```

```
    perror("Unable to set input");
```

```
    goto fatal;
```

```
}
```

在

```
/*
```

```
 * set framerate
```

```
*/
```

前添加

```
vd->fmt.type = 0x80;
```

```
ret = ioctl(vd->fd, VIDIOC_S_FMT, &vd->fmt);
```

```
if (ret < 0) {
```

```
    perror("Unable to set format");
```

```
    goto fatal;
```

```
}
```

注释如下代码

```
/*
```

```
 * set framerate
```

```
*/
```

```
struct v4l2_streamparm *setfps;
```

```
setfps = (struct v4l2_streamparm *) calloc(1, sizeof(struct v4l2_streamparm));
memset(setfps, 0, sizeof(struct v4l2_streamparm));
setfps->type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
setfps->parm.capture.timeperframe.numerator = 1;
setfps->parm.capture.timeperframe.denominator = vd->fps;
ret = ioctl(vd->fd, VIDIOC_S_PARM, setfps);
```

修改

```
vd->buf.length, PROT_READ, MAP_SHARED, vd->fd,
为
vd->buf.length, PROT_READ | PROT_WRITE, MAP_SHARED, vd->fd,
```

修改 int uvcGrab(struct v4l2 *vd)函数

修改

```
ret = ioctl(vd->fd, VIDIOC_DQBUF, &vd->buf);
if (ret < 0) {
    perror("Unable to dequeue buffer");
    goto err;
}
为
retry:
ret = ioctl(vd->fd, VIDIOC_DQBUF, &vd->buf);
if (ret < 0) {
    usleep(10000);
    goto retry;
}
```

修改

```
memcpy(vd->framebuffer, vd->mem[vd->buf.index], (size_t) vd->bytesused);
为
memcpy(vd->framebuffer, vd->mem[vd->buf.index], (size_t) vd->framesizeIn);
```

修改 static int video_enable(struct v4l2 *vd)函数

在

```
vd->isstreaming = 1;
前添加
sleep(1);
```

c) 编译 mjpg-streamer

```
make clean
make
```

编译成功显示如下:

```
arm-cortex_a8-linux-gnueabi-gcc -c -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o huffman.lo hu
arm-cortex_a8-linux-gnueabi-gcc -c -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o marker.lo mar
arm-cortex_a8-linux-gnueabi-gcc -c -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o quant.lo quan
arm-cortex_a8-linux-gnueabi-gcc -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o input_gspcavi1.so
ncoder.lo huffman.lo marker.lo quant.lo
make[1]: Leaving directory `/home/david/Exynos4412/test/mjpg-streamer/mjpg-streamer-r63/plugins/i
cp plugins/input_gspcavi1/input_gspcavi1.so .
```

- d) 拷贝程序及库到根文件系统中

```
$ cp *.so /source/rootfs/mjpg
$ cp mjpg-streamer /source/rootfs/bin
```

3、测试 mjpg-streamer

- a) 在 FS4412 开发板上运行如下指令：

```
mjpg_streamer -i "/mjpg/input_uvc.so -y -d /dev/video0" -o "/mjpg/output_http.so -w
192.168.9.111:8080"
```

执行成功显示如下：

```
MJPEG-streamer [968]: www-folder-path...: 192.168.9.111:8080/

o: HTTP TCP port.....: 8080
MJPEG-streamer [968]: HTTP TCP port.....: 8080

o: username:password.: disabled
MJPEG-streamer [968]: username:password.: disabled

o: commands.....: enabled
MJPEG-streamer [968]: commands.....: enabled

MJPEG-streamer [968]: starting input plugin
MJPEG-streamer [968]: starting output plugin: /mjpg/output http.so (ID: 00)
```

- b) 打开网页输入网址查看摄像头采集到的图形：

浏览器上地址栏输入如下内容：

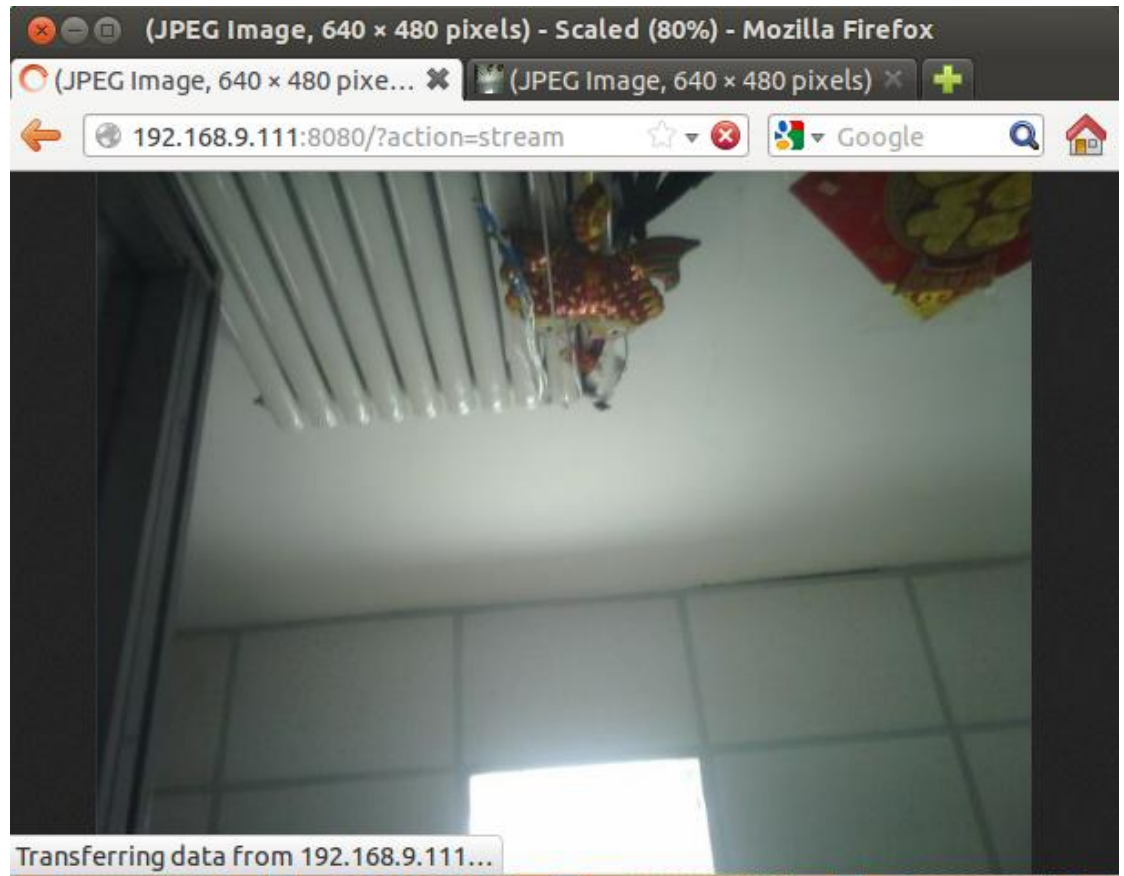
<http://192.168.9.111:8080/?action=snapshot>

或者

<http://192.168.9.111:8080/?action=stream>

snapshot 表示每次抓拍一张图形显示在网页上，stream 表示视频流也就是连续的图像

现象如下：



如果能正常看到图形则表示移植成功，本文前提是 FS4412 的摄像头驱动能够正常使用。